

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh prototypu v rámci vývoje systému pro autoservis
Design of Prototype Within Car Service Information System Development

Student:
Vedoucí bakalářské práce:

Filip Skulník
Ing. Radek Němec, Ph.D.

Ostrava 2019

Zadání bakalářské práce

Student:

Filip Skulník

Studijní program:

B6209 Systémové inženýrství a informatika

Studijní obor:

6209R017 Informatika v ekonomice

Téma:

Návrh prototypu v rámci vývoje informačního systému pro autoservis
Design of a Prototype Within Car Service Information System
Development

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Úvod
2. Teoretická východiska vývoje a nasazení databázových aplikací
3. Analýza současného stavu a definice požadavků
4. Návrh iterace vývoje informačního systému
5. Tvorba a testování prototypu části rozhraní
6. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků bakalářské práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

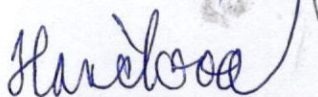
KROENKE, David a David J. AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.
GRÄSSLE, P., H. BAUMANN and P. BAUMANN. *UML 2.0 in Action - a Project Based Tutorial*.
Birmingham: Packt Publishing, 2005. ISBN 978-1-904811-55-8.
PECINOVSKÝ, Rudolf. *Java 9: kompletní příručka jazyka*. Praha: Grada Publishing, 2018.
ISBN 978-80-271-0715-5.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

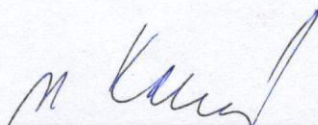
Vedoucí bakalářské práce: **Ing. Radek Němec, Ph.D.**

Datum zadání: 23.11.2018

Datum odevzdání: 10.05.2019



prof. Ing. Jana Hančlová, CSc.
vedoucí katedry



prof. Dr. Ing. Zdeněk Zmeškal
děkan fakulty

Prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně.

V Ostravě dne 10. května 2019

.....

Filip Skulník

Rád bych na tomto místě poděkoval váženému panu Ing. Radku Němcovi, Ph.D. z Katedry systémového inženýrství za to, že vedl mou práci s ochotou, trpělivostí a že mi pomohl v této práci.

Obsah

1	Úvod	5
2	Teoretická východiska tvorby databázových aplikací	7
2.1.	Vysvětlení základních pojmů	7
2.1.1	Data	7
2.1.2	Informace	7
2.1.3	Znalost	8
2.1.4	Databáze	8
2.1.5	Systém řízení báze dat (SŘBD)	9
2.1.6	Databázová aplikace	9
2.1.7	CRM – Customer Relationship Management	10
2.1.8	ERP – Enterprise Resource Planning	11
2.1.9	Funkční požadavky	11
2.1.10	Jazyk UML	12
2.1.11	JAVA	13
2.1.12	MySQL	14
2.1.13	HTML5	17
3	Analýza současného stavu a definice požadavků	20
3.1.	Popis současného stavu	20
3.2.	Nefunkční požadavky	20
3.3.	Funkční požadavky	21
3.3.1	Seznam funkčních požadavků	21
3.3.2	Případy použití	22
3.3.3	Trasovací matice případů použití	30
3.3.4	Přidělování přístupu k systému	31
4	Návrh iterace vývoje informačního systému	32
4.1.	Konceptuální model databáze systému	32
4.1.1	Entita Zákazník	33
4.1.2	Entita Objednávky	34
4.1.3	Entita Dodací listy	34
4.1.4	Entita Faktura	34
4.1.5	Entita Platby	34
4.1.6	Entita platební podmínky	35
4.1.7	Entita Auta	35

4.1.8	Entita Kola	35
4.1.9	Entity Položky a Seznam položek	35
4.2.	Návrh logické struktury databáze	36
4.2.1	Tabulka customer	36
4.2.2	Tabulky transakcí	36
4.2.3	Tabulka login	37
5	Tvorba a testování prototypu	38
5.1.	Návrh grafické uživatelského rozhraní	40
5.2.	Hlavní prvky rozhraní (stránky) webové aplikace	42
5.2.1	Nástěnka	42
5.2.2	Zákazníci	42
5.2.3	Portfolio zákazníka	43
6	Závěr	44

1 Úvod

Tématem této bakalářské práce je návrh jednoduché databázové aplikace a tvorba prototypu pro autoservis ve městě Prostějov, v Olomouckém kraji. Důvod pro vytvoření aplikace je modernizace vedení záznamů o jednotlivých zakázkách, pohledávkách, kolech na skladě, a hlavně vedení záznamů o stálých zákaznících.

V současnosti jsou zakázky vedeny na zakázkových arších, které slouží jako podklad pro fakturaci.

V servise se používá jednoduchý a zastaralý informační systém, který nedokáže zaznamenávat informace o zákaznících, jejich uskladněných kolech, jejich autech a pohledávkách jimi vytvořených.

Záznamy o kolech na skladě jsou vedeny v souboru Microsoft Excel, který se jednou za půl roku obměňuje, jelikož během sezóny přezouvaní se mnoho pneumatik vyhodí a někteří zákazníci třeba dále uskladňovat pneumatiky nechtějí.

Cílem je navrhnout jednoduchý informační systém a vytvořit prototyp, který bude umožňovat vést záznamy o jednotlivých zakázkách, pohledávkách a položkách na skladě. Aplikace by měla:

- zaznamenávat data o stálých zákaznících,
- zaznamenávat pohledávky,
- zaznamenávat položky na skladě,
- mít jednoduchý vzhled a ovládání.

Koncept této bakalářské práce je tvořen z šesti částí.

V další části „Teoretická východiska vývoje a nasazení databázových aplikací“, jsou popsány základní pojmy problematiky databázových aplikací, a seznámení s HTML (HyperText Markup Language) s objektově orientovaným programovacím jazykem Java 9 a systémem řízení báze dat MySQL, pomocí kterých by měla být aplikace vytvořena.

Třetí částí je „Analýza současného stavu a definice požadavků“, kde bude vystižen současný stav zakázkového systému v autoservise a definovány požadavky na nový informační systém.

Čtvrtou částí je část „Návrh iterace informačního systému“, která se bude zabývat návrhem iterace v rámci procesu vývoje databázové aplikace, tak aby splňoval požadavky autoservisu a vytvářel přidanou hodnotu pro servis pomocí formulářů, přehledů a sestav.

V paté části „Tvorba a testování prototypu části rozhraní“ bude popsána tvorba prototypu, který se sestává z pouze vizuální stránky aplikace tzv. frontendu jako návrh grafického uživatelského rozhraní.

Poslední částí je „Závěr“, kde bude zhodnocení návrhu a prototypu v autoservise. Návrhy na zlepšení a rozšíření funkcionalit systému do budoucnosti.

2 Teoretická východiska tvorby databázových aplikací

V informatických oborech se používají různé technologie a pojmy, které je třeba znát a umět používat. V informatice se tyto technologie neustále vyvíjejí a je nutné zůstat v obraze.

2.1. Vysvětlení základních pojmů

Tahle podkapitola se bude věnovat vysvětlení základních pojmů a to jsou:

- Data,
- Informace,
- Znalost,
- Databáze,
- Systém řízení báze dat,
- Databázová aplikace.

Tyhle pojmy jsou klíčové pro pochopení problematiky a měl by je znát každý, kdo chce dané problematice porozumět. Dále se tato kapitola věnuje technologiím pro návrh informačních systémů a důležitým pojmům, které se týkají informatiky pro podnik.

2.1.1 Data

Data je možné rozlišovat na:

Strukturovaná – mají logickou datovou strukturu, díky které se s těmito daty lépe pracuje. Například datová struktura objekt má své atributy, datové položky a metody. Díky logickým vazbám mezi těmito daty lze vybírat jen ta data, která jsou potřeba.

Nestrukturovaná – se berou jako datový tok bitů, které reprezentují uloženou položku v počítači jako jsou videa, zvukové stopy, textové dokumenty atd. [2]

2.1.2 Informace

Pojem Informace z latinského in-formatio, znamená v překladu: ztvárnění, utváření. Informace jsou data nesoucí nějaký význam získán výkladem určitých dat.

Pro pojem „informace“ existuje mnoho definic, protože je to velmi široký pojem.

Další definice informací:

Informace jsou zpracovaná data, která získala takovou strukturu, že jsou interpretovatelné pro daný subjekt, kterému poskytuje přidanou hodnotu. [2]

2.1.3 Znalost

Znalosti člověk nabývá studiem a zpracováním informací tím, že informace porovnává, třídí, zasazuje do kontextu ostatních informací, vyhodnocuje důsledky, a tak vytváří strukturu informací.[3]

2.1.4 Databáze

Databáze je organizovaná kolekce dat, obecně uložených a přístupných elektronicky skrze počítačové systémy. Databáze jsou používány pro ukládání, řízení a vrácení informací, které jsou pro daného uživatele užitečné. Velmi běžný je databázový koncept relačních databází jako Oracle, SQL Server a MySQL, kde jsou typicky data uloženy v tabulkovém formátu. [3]

Databáze jsou definovány také jako sebe popisující kolekce integrovaných záznamů. Popis dat je znám jako systémový katalog či metadata (data o datech). Přístup k databázovým systémům, kde je definice dat separovaná od aplikačních programů, je podobný přístupu v moderním vývoji softwaru, kde je vnitřní definice objektu oddělená od vnější definice. Uživatelé vidí pouze vnější definici a vůbec netuší, jak je objekt definován a jak funguje. Jedna výhoda tohoto přístupu známého jako abstrakce dat, je že můžeme měnit vnitřní definici objektu bez postihnutí uživatele objektu a bez změny vnější definice. Jestliže přibudou nová data, stará data se modifikují. [3]

Další definice databáze je logicky vztažená. Když se analyzují informace o potřebách organizace, je snaha o identifikaci entit, atributů a vztahů. Entita je zřetelný objekt (osoba, místo, věc, koncept nebo událost) v organizaci, který by měl být v databázi reprezentován. Atribut je vlastnost, která popisuje nějaký aspekt objektu, který chceme zaznamenat a vztah je asociace mezi entitami. Pro znázornění této logické vztažnosti se používá „Entity Relationship Diagram“. [3]

Databázové architektury používají programovací jazyky pro návrh konkrétního typu softwaru pro byznys či jiné organizace. Databázová architektura se zaměřuje na návrh, vývoj, implementaci a údržbu počítačových programů, které ukládají a organizují informace pro byznys, agentury a instituce. Databázová architektura věnuje vývoji a implementaci softwaru podle požadavků uživatele. [3]

2.1.5 Systém řízení báze dat (SŘBD)

Systém řízení báze dat je software, který interaguje s uživatelskými programy a databázemi. Typicky, SŘBD poskytuje následující možnosti:

- Umožňuje uživatelům definovat databázi, běžně skrze „Data Definition Language“ (DDL), který umožňuje uživatelům specifikovat typy dat, struktury a omezení nad daty, které jsou uloženy v databázi. [3]
- Umožňuje uživatelům vkládat, měnit a vracet data z databáze, obvykle skrze „Data Manipulation Language“ (DML). Mít centrální úložiště pro všechna data a popisy dat umožňuje DML poskytovat hlavní dotazovací nástroj pro tyto data, který se nazývá query language (dotazovací jazyk). Nejběžnější dotazovací jazyk je „Structured Query Language“ (strukturovaný dotazovací jazyk), který je dnes jak formálním, tak i defacto standardním jazykem pro komunikaci s relačním SŘBD. [3]
- Poskytuje kontrolovaný přístup k databázi. Poskytuje například:
 - systém zabezpečení, který zabraňuje neautorizovaným uživatelům přístup do databáze;
 - systém integrity, který má na starosti údržbu konzistence uložených dat;
 - systém kontroly souběhu operací nad daty (Concurrency control system), který umožňuje sdílený přístup do databáze;
 - systém kontroly zálohování, který obnoví databázi do předchozího konzistentního stavu následně po hardwarovému či softwarovému selhání,
 - a uživatelsky přístupný katalog, který obsahuje popisy dat v databázi.

[3]

2.1.6 Databázová aplikace

Databázová aplikace je počítačový program, který operuje s databázemi pomocí SQL dotazů na SŘBD za účelem zjistit z databází informace, které uživatel žádá. Databázová aplikace je psána ve vyšších programovacích jazycích. [2]

Uživatelé interagují s databázemi skrze mnoho aplikačních programů, které jsou používány pro vytváření a údržbu databází a pro generování informací. Tyto programy mohou být konvenční aplikace v počítači nebo dnes více typické online aplikace. [2]

Databázové aplikace jsou používány pro vyhledávání, třídění, řízení a sdílení informací. Databázové aplikace poskytují bezpečnost omezením přístupu k datům autorizací pomocí přihlašovacích údajů a hesel. [3]

Mnohé moderní webové stránky používají více databázových aplikací naráz jak jádrové komponenty. Většina internetových obchodů používá databáze pro ukládání dat o tržbách. Tyto stránky taktéž kombinují účetní databázové systémy pro zaznamenání transakcí z tržeb a CRM (Customer Relationship Management) databázové aplikace, pro poskytnutí zpětné vazby a řízení pozitivní zkušenosti zákazníka. [3]

2.1.7 CRM – Customer Relationship Management

„Customer Relationship Management“ neboli „Řízení zákaznických vztahů“ je strategická komponenta řízení byznysu, jejímž cílem je řídit vztahy a interakce organizace se zákazníky a potenciálními zákazníky. CRM systém pomáhá společnostem zůstat ve spojení se zákazníky, usměrnit procesy a zlepšit výnosnost. [3]

Cílem CRM je vylepšit vztahy podniku se zákazníky.

Jak porozumět CRM:

- CRM jako technologie: Technologický produkt, který se používá pro zaznamenání, hlášení a analýzu interakcí zákazníka a podniku. Řízení dat o zákaznících a statistiky jejich akcí.
- CRM jako strategie: Strategie podniku, která říká, jak by se měly řídit vztahy se zákazníky a potenciálními zákazníky
- CRM jako proces: Proces v podniku, který se skládá z jednotlivých úkonů, které zabezpečují zdravé vztahy se zákazníky.

CRM software vede záznamy o kontaktních informacích zákazníka jako jsou: telefonní číslo, email, adresy (doručovací a fakturační). Systém CRM organizuje data o zákaznících, tak aby byli vidět širší souvislosti pro porozumění vztahu postupem času.

2.1.8 ERP – Enterprise Resource Planning

ERP (Enterprise resource planning) neboli „Plánování podnikových zdrojů“ je integrované řízení hlavních podnikových procesů, často v reálném čase prostřednictvím softwaru a technologií. [7]

ERP je běžně bráno jako kategorie byznys-management softwaru – typicky jako balíček integrovaných aplikací, které organizace používá pro sběr, ukládání, řízení a interpretace dat z mnohých podnikových aktivit. [7]

ERP software nabízí určité stupně synchronizovaných zpráv a automatizaci. Místo toho, aby se zaměstnanci starali o separované databáze a tabulky, které musí manuálně spojit pro generování hlášení, některé ERP řešení umožňuje zaměstnancům vytáhnout zprávy z jednoho systému. Například objednávky odběratelů automaticky přejdou do finančního systému bez manuálního rozklíčování, sekce řízení objednávek může vyřizovat objednávky rychleji a snadněji a finanční sekce uzavře knihy dříve. [7]

ERP zastřešuje sekce podniku napříč celým podnikem. ERP zahrnuje účetnictví, řízení zakázek, řízení projektů, řízení rizik a operace s dodávkami. ERP má velký význam pro vrcholové řízení podniku v plánování, rozpočtech predikci a zprávách o finančních výsledcích podniku. [7]

Přidaná hodnota ERP se v podniku projevuje zlepšením vhlížení do byznysu, snížením operačních nákladů, zlepšení kooperace mezi uživateli, konzistentní infrastrukturou, snížením rizik finančními kontrolami. [7]

2.1.9 Funkční požadavky

Funkční požadavky definují funkce systému nebo jeho komponent, kde je popsána jako specifikace chování mezi výstupy a vstupy. Funkční požadavky mohou zahrnovat kalkulace, technické detaily, procesy a manipulace s daty a jiné specifické funkcionality, které definují, čeho má systém dosáhnout. Požadavky chování systému popisující všechny případy, kde systém využívá funkční požadavky, jsou zachyceny v případech použití tzv. Use cases. [3]

2.1.10 Jazyk UML

UML (Unified Modeling Language) je soubor diagramů, grafických notací, který je standardem pro analýzy a návrhy informačních systémů. Pro programátory je důležité, aby se v těchto diagramech orientovali. UML se používá v dokumentacích návrhů IS jako nástroj pro usnadnění vývoje. UML lze použít i pro byznys modelování a modelování nesoftwarových systémů. [1]

V historii objektově-orientovaného programování se hledaly způsoby z jakých všemožných úhlů se dívat na IS. Postupně vznikaly různé specifikace a standardy pro zakreslování objektů a dalších součástí aplikací. V devadesátých letech firma Rational Software sjednotila několik metodik a vytvořila standard UML. Postupem času vznikala poptávka o tento standard a existovala touha UML udržet a zdokonalovat. Díky standardizaci v softwarových společnostech se UML uchytilo, protože zjednodušuje komunikaci, a tudíž se programátoři nemuseli učit další jazyky pro další projekty a snížily se tak náklady na školení. Pro jazyk UML vzniklo konsorcium OMG (Object Management Group), otevřená organizace, která dohlíží na rozvoj specifikace UML. V OMG působí firmy jako IBM, Oracle, Rational Software a další, kteří společně vyvíjí tento modelovací jazyk. [1]

Modelování je navrhování softwaru před programováním. Modelování je nezbytná část velkých softwarových projektů a velmi užitečné u středně velkých i malých projektů. Model hraje důležitou roli v softwarovém vývoji, tak jako technické kresby a další plány hrají ve stavění velkých budov. Zodpovědní, za úspěch ve vývoji softwarových projektů, se mohou modely ujistit, že funkcionality byznysu jsou kompletní a správné, že potřeby koncového uživatele jsou naplněny a návrh programu podporuje požadavky na škálovatelnost, robustnost, bezpečnost a další charakteristiky, před samotnou implementací, jelikož potom by změny byly dosti drahé. Velké projekty mají vysokou pravděpodobnost selhání. Fakticky vzato je větší pravděpodobnost, že velké projekty selžou, v naplnění všech požadavků, potřeb byznysu a dodržení rozpočtu, než že uspějí. [1]

2.1.10.1 Diagramy UML

UML 2.0 definuje třináct typů diagramů rozdělených do tří kategorií. Šest diagramů pro znázornění statické struktury aplikace, tři pro znázornění obecného chování aplikace a čtyři reprezentující různé aspekty interakcí v aplikaci. [11]

- **Strukturní diagramy:** diagram tříd (Class diagram), diagram objektů, diagram komponent, diagram kompozice struktury, diagram balíčků (Package diagram), diagram rozestavění. [11]
- **Diagramy chování:** diagram případů použití (Use case diagram), Activity diagram a state machine diagram (stavový diagram). [11]
- **Diagramy interakcí:** diagram chování (z něho pak byly derivovány další diagramy), sekvenční diagram, komunikační diagram, časový diagram a diagram přehledu interakcí. [11]

2.1.11 JAVA

Java je objektově-orientovaný jazyk, jehož hlavní výhodou je přenositelnost a jednoduchost. Je to nezávislý, bezpečný, výkonný, robustní, distribuovaný, dynamický, mnohováknový a populární programovací jazyk. Za velkou oblíbenost Javy může právě přenositelnost a jednoduchost. Jazyk Java byl vyvinut společností Sun Microsystems, kde James Gosling, Bill Joy, Mike Sheridan a Patrick Naughton započali projekt s cílem vyvinout systém pro domácí spotřebiče. Projekt nejprve měl název „Stealth project“, ale následně byl přejmenován na „Green project“. V projektu, tzv. Green team, nejprve používal programovací jazyk C++, který ale nevyhovoval cíli. James Gosling proto začal pracovat na novém jazyku s názvem Oak(dub) podle stromu, jež stál pod okny kanceláře. V roce 1995 byl Oak přejmenován na Java (přeloženo z amerického slangu na „kafe“), protože jazyk Oak již existoval. [4]

Java je i platforma, která zahrnuje soubor programů pro vývoj a spouštění aplikací. Funkce Javy je utvořena pro různé operační systémy a hardwarové komponenty. Hlavní platformy Javy jsou [4]:

- Java Card – platforma pro provoz aplikací na tzv. Smart cards (bankovní karty a podobně) nebo na dalších malých zařízeních.
- Java Micro Edition – podmnožina platformy Java SE (Standard Edition) s nabídkou sbírek API (API – Application Programming Interface, aplikační programové rozhraní), určených pro vývoj softwaru na malých zařízeních s omezenými prostředky (mobilní zařízení, telefony, pagery atd.).

- Java SE (Standard Edition) – je platforma Javy, která byla vyvíjena od první verze. Jedná se o původní kolekci API pro aplikace na stolní počítače a pro odlišení od ostatních platforem se zavedlo toto označení.
- Java EE (Enterprise Edition) – platforma pro vývoj a provoz rozsáhlých informačních systémů.

Syntaxi jazyka, virtuální stroj JVM (Java Virtual Machine) a API tyto platformy sdílejí. Na rozdíl od jazyka C/C++ není třeba explicitně alokovat paměť a následně uvolňovat, protože Java Virtual Machine obsahuje program zvaný Garbage collector (sběrač odpadu), který se o operační paměť programu stará tak, aby nedošlo k vyčerpání paměťových zdrojů či přetečení paměti. [4]

2.1.12 MySQL

MySQL je světově nejoblíbenější open source (s veřejným zdrojem) relační SŘBD. S její osvědčenou výkonností, spolehlivostí a jednoduchým použitím se MySQL stala vedoucí volbou databáze pro online databázové aplikace, mimo jiné vysoce používaná weby jako jsou Facebook, Youtube, Twitter a další. [6]

MySQL byla vytvořena společností MySQL AB, kterou založili David Axmark, Allan Larsson a Michael Widenius. První verze MySQL se objevila 23.května v roce 1995. Zprvu se jednalo o SŘBD pro personální použití na jazyku ISAM, který vývojáři zavrhlí poté, co zjistili, že je moc pomalý a nepružný. Proto vytvořili nové SQL rozhraní s tím, že zachovají API z předešlé verze. [6]

V roce 2008 firma Sun Microsystems koupila MySQL AB a v roce 2010 byla Sun Microsystems koupena firmou Oracle, a tudíž MySQL dnes spravuje Oracle. [6]

Struktury v MySQL jsou organizovány ve fyzicky oddělených souborech optimalizovaných pro rychlost. Logický model, s objekty jako jsou databáze, tabulky, přehledy, řádky a sloupce, nabízí flexibilní programovací prostředí. SŘBD vynucuje tyto pravidla, takže dobře navrhnutá databáze nemůže být nekonzistentní, obsahovat duplikace nebo mít chybějící data. [6]

2.1.12.1 Hlavní funkce

MySQL je napsáno v programovacím jazyce C a C++. Pro přenosnost používá open source platformu Cmake ve verzích od 5.5 a dál. Používá vícevrstvé návrhy serverů s nezávislými moduly. Je navrhnuté jako multi-vláknové k jednoduchému užití dostupných CPU (Central processing unit) a dynamické alokaci dostupné paměti. Poskytuje transakční i netransakční úložní engine. Poskytuje server jako separátní program, pro použití v klient/server síťovém prostředí, a jako knihovnu připojenou k samostatným aplikacím. [6]

2.1.12.2 Datové typy

Datové typy pro čísla [6]:

- Celočíselné: TINYINT, SMALLINT, MEDIUMINT, INT a BIGINT.
- S desetinou čárkou: DECIMAL, FLOAT, DOUBLE.
- Pole bitů: BIT.

Datové typy pro řetězce [6]:

- Znakové: CHAR, VARCHAR, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT.
- Binární: BINARY, VARBINARY, BLOB (Binary Large Object, binární soubor), TINYBLOB, MEDIUMBLOB, LARGEBLOB.
- Výčet: ENUM.

Datové typy pro čas: DATE, TIME, DATETIME, TIMESTAMP, YEAR.

2.1.12.3 Výrazy a funkce

Příkazy pro manipulaci s daty [6]:

- SELECT – výběr dat z databáze a jejich řazení.
- INSERT – vkládání dat do databáze.
- UPDATE – změny dat v databázi.
- MERGE – jestli existuje odpovídající klíč, data změní, pokud ne, tak je vloží.
- DELETE – mazání dat.
- EXPLAIN – zobrazení postupu zpracování SQL příkazu.

- SHOW – zobrazení vnitřních definic dat.

Příkazy pro definici dat [6]:

- CREATE – vytvoření objektů.
- ALTER – změny objektů.
- DROP – Odstranění objektů.

Příkazy pro řízení dat [6]:

- GRANT – pro přidělení práv uživateli k daným objektům.
- REVOKE – pro odstranění práv uživatele.
- START TRANSACTION – zahajování transakce.
- COMMIT – potvrzení transakce.
- ROLLBACK – zrušení transakce (při zrušení transakce se databáze vrátí do konzistentního stavu před transakcí).

2.1.12.4 Objekty SQL

Objekty reprezentují tabulky, pohledy, index, omezení, transakce a spouště. [6]

- TABLE – tabulka databáze, definují se sloupce a jejich atributy.
- VIEW – pohled ve formátu tabulky. Data se zobrazí podle předpisu, jakým způsobem se mají data uživateli zobrazit.
- INDEX – databázová konstrukce, která slouží pro rychlejší vyhledávání.
- CONSTRAINT – omezení pro vkládání, změnu a odstranění dat v tabulkách.
- TRANSACTION – sada příkazů pro manipulaci s daty, převádí databázi z jednoho konzistentního stavu do druhého. Pokud nebude druhý stav konzistentní, příkaz ROLLBACK databázi uvede do předešlého konzistentního stavu.
- TRIGGER – spouštěč reakcí na akci definovanou při vytváření tohoto objektu.

2.1.13 HTML5

HTML5 je hypertextový značkovací jazyk, kterým se vytvářejí webové stránky. Pro tvorbu webových stránek se používají tři nezávislé komponenty: HTML, který udává strukturu stránky, kaskádové styly (CSS – Cascading Style Sheets), které se starají o stylizaci a prezentaci obsahu, a JavaScript, který uvádí věci do chodu. [5]

2.1.13.1 Struktura HTML

HTML kód se skládá ze značek ve špičatých závorkách pro oddělení klíčových slov dokumentu od prostého textu. Značky HTML mohou být párové či nepárové, některé mohou mít atributy a některé nesmí. Párové značky mají tento formát `<...> </...>`. Značky neboli tagy se používají v dokumentu pro: formátování, tabulky, odkazy, bloky, seznamy, formuláře, obrázky, klikací mapy, multimedia, rámy a další. [5]

Každý dokument HTML začíná párovou značkou `<html>` a končí `</html>`. Na začátku dokumentu ještě před značkou `<html>` se definuje tzv. doctype, který uvádí specifikaci dokumentu. Doctype je důležitý pro webové prohlížeče, protože podle doctypeu určí, jak se má stránka vykreslovat. [5]

Po značce `<html>` následuje párová značka `<head>`, která vymezuje hlavičku dokumentu, kde jsou uvedeny metadata o stránce jako titulek dokumentu, znaková sada, styly, odkazy na externí skripty a další metadata. Značky metadat jsou: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>` a `<base>`. [5]

Po hlavičce dokumentu následuje tělo dokumentu vymezeno párovou značkou `<body>`. Do těla dokumentu se píše kód obsahu dokumentu, ve kterém se mohou vyskytovat obrázky, text, videa, odkazy, formuláře, tabulky a další věci. [5]

2.1.13.2 Kaskádové styly

Kaskádové styly jsou kolekcí metod pro definici grafické úpravy webových stránek. Kaskádové styly se mohou na sebe vrstvit (proto kaskádové), ale platit bude pouze ta poslední vrstva. [5]

Kaskádové styly formátují typ písma, barvu písma, text, odstavce, zalamování řádků, odrážky, číslování a také tabulky a rámečky atd. [5]

CSS se dá použít třemi způsoby [5]:

- Přímě v těle formátovaného elementu za použití atributu *style*="..."
- V hlavičce pomocí párového tagu `<style></style>`, do jehož těla se zapíše, jaký element má mít jaké formátování.
- Pomocí připojení externího souboru s koncovkou *.css, ve kterém jsou jednotlivé styly elementů definovány. Soubor se připojí pomocí tagu `<link>`, do něhož se pomocí atributu *href*, ve kterém je relativní adresa k onomu souboru kaskádových stylů.

2.1.13.3 JavaScript

JavaScript je programovací jazyk webových stránek, umožňující zahrnout aktivní prvky do původně statických a samotný program se vykonává až na straně klienta v prohlížeči-

Autor tohoto jazyka je Brendan Eich, který ho vyvinul v roce 1995, když pracoval ve firmě Netscape. Zprvu se tento jazyk nazýval Mocha, následně byl přejmenován na LiveScript. Nakonec dostal jméno JavaScript z marketingových důvodů. Asi proto, že v roce 1995 byl uveden jako doplněk k jazykům Java a HTML. [8]

JavaScript je jazyk interpretovaný, objektově-orientovaný a závislý na prohlížeči, takže funguje pouze v prohlížeči a spouštění programu je typicky řízeno událostmi, kterým dochází při interakci uživatele s webovou stránkou ve webovém prohlížeči. V JavaScriptu záleží na velikosti písma a má podobnou syntaxi jako C a Java, přičemž Java a JavaScript jsou dva úplně rozdílné jazyky. Jelikož existují různé verze jazyka a prohlížečů, tak dochází často k chybám. [8]

JavaScript se používá často také k ovládání grafického uživatelského rozhraní. Dají se s ním tvořit různé animace a obrázkové efekty.

Kód v jazyce JavaScript lze do kódu HTML dokumentu vložit třemi způsoby [8]:

- za použití párového tagu `<script>` a to kamkoliv do proudu dokumentu,
- odkazem na externí soubor s koncovkou *.js,
- a třetí možnost je bez tagu, zapsán v řádku jako atribut nějakého tagu.

Prakticky e kód v jazyce JavaScript spouští dvěma způsoby [8]:

- ihned při zavádění stránky
- jako reakce na nějakou uživatelskou akci (najetím kurzoru na HTML element, který je oskriptován)

3 Analýza současného stavu a definice požadavků

3.1. Popis současného stavu

Požadavky na nový informační systém vycházejí ze stávajícího stavu v autoservise. Jelikož je užíván starý informační systém pro fakturování, příjem zboží, výdej zboží, a správu položek na skladě, je třeba modernizovat a to tak, že nový systém bude poskytovat informace o zákazníkovi a jeho akcích v pneuservise.

Objednávky se vyřizují telefonicky, nebo emailem a zapisují se do objednávkové knihy, ve které se jednotlivé objednávky po vyřízení a vyfakturování přeškrtnou jednou čarou. Z knihy nelze nijak vytahovat jednoduše informace o jednotlivých zákaznících a jejich objednávkách, proto je potřeba objednávky vést v elektronické podobě pomocí aplikace, která uchovaná data v databázi dokáže zpracovat a poskytnout výstupy pro zkvalitnění služeb servisu. Záznamy o zákaznících jsou dohledatelné pouze z vydaných faktur, přeškrtnutých objednávek a informace o uskladnění kol jsou vedené v souboru Microsoft Excel.

Stávající informační systém je zastaralý, nevede záznamy o zákaznících, jejich objednávkách, jejich uschovaných kolech na skladě a jejich závazcích. Je umístěn na jednom počítači, který když přestane fungovat, tak bude servis mimo provoz do doby, než se z něj podaří vytáhnout potřebná data. Je nepřenositelný a nemá cloudovou zálohu. Záměrem je tedy, aby systém byl uložen na serveru a byl přístupný přes prohlížeč.

3.2. Nefunkční požadavky

Nefunkční požadavky definují atributy systému jako jsou: bezpečnost, výkonost, udržitelnost, škálovatelnost a použitelnost. [3]

Uživatelské rozhraní tohoto systému by měla být jednoduché a mělo by posílat požadavky klienta na server, kde program napsaný v Javě vyhodnotí požadavek a pošle na stranu klienta požadovanou stránku s požadovaným obsahem. Program na straně serveru bude komunikovat s databází pomocí SQL příkazů a dotazů.

Do webové aplikace se uživatelé nebudou moci registrovat. Jejich registraci může provést jen oprávněná osoba nebo administrátor. Po prvním přihlášení si budou muset změnit heslo.

3.3. Funkční požadavky

Pro zpracování funkčních požadavků jsem použil tzv. Případů použití (Use cases), což je velmi používaný a oblíbený nástroj pro popsání funkcí informačního systému.

V systémovém inženýrství jsou případy použití seznamem akcí a událostí typicky definující interakce, mezi rolemi (v UML aktéři) a systémem, tak aby bylo dosaženo cíle. Aktér může být člověk, který přichází s daným systémem do styku, či jakýkoli externí systém. [11]

3.3.1 Seznam funkčních požadavků

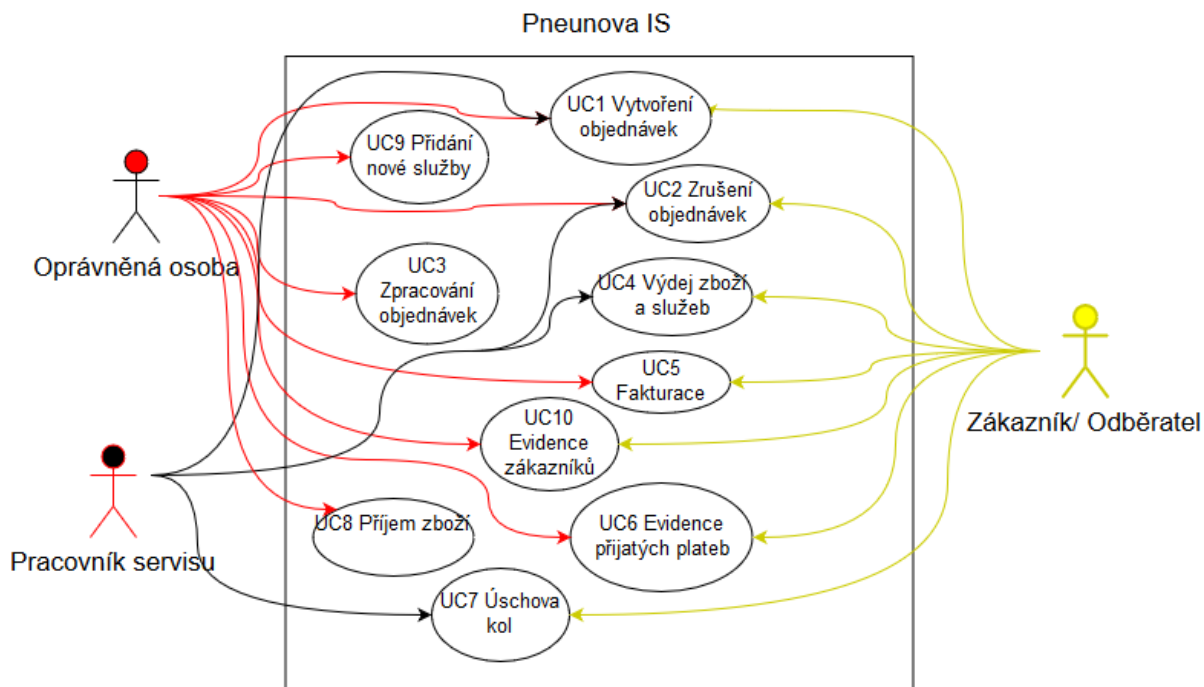
Tyto požadavky jsou zpracovány pomocí případů použití. Označení funkčních požadavků slouží pro trasování případů použití, které se řeší trasovací maticí (viz tabulka 1).

Systém bude:

- R1 – spravovat objednávky.
- R2 – spravovat dodací listy.
- R3 – spravovat faktury.
- R4 – párovat platby s pohledávkami.
- R5 – vést záznamy o zboží a službách nabídky.
- R6 – vést záznamy o úschově zákazníků.
- R7 – vést záznamy o zákazníkovi.

3.3.2 Případy použití

Případy použití níže, popisují procesy v servise, v něž figurují aktéři, kteří nějakým způsobem interagují se systémem.



Obrázek 1 - Diagram případů použití

3.3.2.1 UC1 Vytvoření objednávek

Aktéři: Zákazník, pracovník servisu. **Cíl:** Vytvořit objednávku

Vstupní podmínky: žádné.

Tok událostí: UC1 začíná požadavkem zákazníka objednat si zboží či službu.

Scénář – Proces vytvoření objednávky:

- Zákazník servisu chce objednat specifické zboží či službu.
- Pracovník servisu na příjmu prověří, zda je dané zboží dostupné na skladě.
- Systém poskytne informaci, že na skladě není žádné dostupné zboží. Pracovník servisu otevře formulář pro vytvoření zákaznické objednávky a vybere existující instanci evidovaného zákazníka nebo vytvoří nový záznam o zákazníkovi (viz UC8).
- Systém automaticky ve formuláři objednávky vyplní jeho údaje (jméno, adresa atd).
- Pracovník servisu na formuláři objednávky v sekci předmět objednávky запиše požadované úkony a zboží stiskne uložit.
- Pokud je zákazník neznámý, pracovník servisu na příjmu jej požádá o adekvátní zálohu k dané objednávce, aby měl záruku, že si zboží vyzvedne.
- Pracovník servisu v objednávkovém formuláři запиše do poznámky zálohu.
- Pracovník servisu vytiskne objednávku a nechá ji zákazníkem podepsat.
- Systém vytvoří nový záznam o objednávce.

3.3.2.2 UC2 Zrušení objednávek

Aktéři: Zákazník, pracovník servisu, oprávněná osoba. **Cíl:** Zrušit objednávku

Vstupní podmínky: Objedávka není vyřízená.

Tok událostí: UC2 začíná požadavkem zákazníka zrušit objednávku.

Scénář – Proces zrušení objednávky:

- Zákazník chce zrušit objednávku. Pokud objednávka ještě nebyla zpracována, lze zrušit.
- Pracovník servisu vyhledá v systému objednávku v seznamu objednávek daného zákazníka. Pokud v poznámce objednávky byla napsána záloha, bude zákazníkovi vrácena.
- Otevře objednávku a dá „Zrušit“.

- Systém objednávku vymaže.

3.3.2.3 UC3 Zpracování objednávek

Aktéři: Oprávněná osoba. **Cíl:** Zpracování objednávky.

Vstupní podmínky: UC1

Tok událostí: UC3 začíná po vytvoření objednávky.

Scénář – Zpracování objednávky:

- Oprávněná osoba otevře v sestavu objednávek, které jsou nevyřízené a postupně je otevírá.
- Otevře objednávku a zjistí, co je potřeba objednat u dodavatele a co je na skladě.
- Do objednávky dopíše, kódy zboží a název zboží, co objednal, nebo položky vloží ze seznamu položek, které již v minulosti byly přijaty.
- Objednávku vytiskne a uloží jako „v procesu“.
- Po dodání zboží dodavatelem, přijme zboží na sklad (viz UC7)

3.3.2.4 UC4 Výdej zboží a služeb

Aktéři: Pracovník servisu, zákazník. **Cíl:** Vydání zboží na dodací list

Vstupní podmínky: zboží je na skladě.

Tok událostí: UC4 začíná, když přijde zákazník pro zboží, nebo jeho objednávka je vyřízená.

Scénář 1 - Výdej zboží odběrateli bez objednávky:

- Odběratel přijde do provozovny a vybere si zboží, které chce
- Pracovník v menu v záložce „Dodací listy“ otevře formulář pro dodací listy.
- Pracovník ve formuláři vybere zákazníka a položky, jež chce vydat.
- Systém vytiskne dodací list a uloží dodací list a jeho předměty do „otevřené“ faktury daného zákazníka nebo otevře novou fakturu a tu uloží jako „otevřenou“.

Scénář 2 - Výdej zboží odběrateli na objednávku:

- Pracovník servisu vyhledá danou zpracovanou objednávku a otevře ji.
- Pracovník vytvoří dodací list akcí „Vytvořit dodací list“ v objednávce.
- Systém otevře nový formulář pro vytváření dodacích listů a zkopíruje hlavičku z objednávky. Pracovník vybere dané zboží pro výdej.
- Systém zkopíruje hlavičku objednávky a naplní dodací list zbožím a službami, které pracovník vybral.
- Systém vytiskne dodací list a uloží dodací list a jeho předměty do otevřené faktury daného zákazníka nebo otevře novou fakturu a tu uloží jako „otevřenou“.

Scénář 3 - Anonymní zákazník odebírá zboží či službu:

- Zákazník přijde do prodejny a požaduje danou službu a zboží.
- Pracovník servisu vytvoří novou fakturu. Naplní fakturu službami a zbožím, které chce fakturovat.
- Jednotlivé položky se, přidáním na fakturu, budou odečítat z evidence skladu.
- Ve formuláři nové faktury zvolí akci „Platit hotově“.
- Systém vytvoří platbu s variabilní symbolem dané faktury.
- Oprávněná osoba přijme hotovost a vytiskne fakturu.
- Systém eviduje fakturu jako „Faktura za hotové“ a jako „Uhrazenou“.

3.3.2.5 UC5 Fakturace

Aktéři: Oprávněná osoba, zákazník. **Cíl:** Vytvořit fakturu.

Vstupní podmínky: Zboží bylo dodáno.

Tok událostí: UC5 začíná v momentu, kdy zboží bylo dodáno a zákazník si přebírá daňový doklad.

Scénář 1 – Faktura za hotové:

- Oprávněná osoba v systému otevře formulář pro novou fakturu.

- Vybere zákazníka a systém vyplní hlavičku. Způsob úhrady zvolí „Hotově“.
- Naplní fakturu službami a zbožím, které chce fakturovat.
- Jednotlivé položky se, přidáním na fakturu, budou odečítat z evidence skladu.
- Systém eviduje fakturu jako „Faktura zaplacenou hotově“.
- Oprávněná osoba fakturu vytiskne a předá zákazníkovi.

Scénář 2 – Faktura se splatností:

- Odběratelská firma chce fakturovat dodací listy na konci měsíce.
- Oprávněná osoba vytvoří novou fakturu, kde zvolí způsob úhrady „Převodním příkazem“ a zvolí možnost „Klasická faktura“.
- Oprávněná osoba naplní fakturu službami a zbožím, které chce fakturovat.
- Jednotlivé položky se, přidáním do faktury, budou odečítat z evidence skladu.
- „Klasická faktura“ je vždy vedena jako „Faktura se splatností“ a dokud k ní nebyla přiřazena platba je evidována jako „Neuhrazená“.
- Oprávněná osoba fakturu vytiskne a předá zákazníkovi.

Scénář 3 – Měsíční faktura:

- Odběratelská firma chce fakturovat dodací listy na konci měsíce
- Oprávněná osoba vytvoří novou fakturu, kde zvolí způsob úhrady „Převodním příkazem“ a zvolí možnost „Měsíční faktura“
- Systém fakturu uloží jako předpřipravenou celý měsíc a je možno do ní přidávat dodací listy, které byly vydány danému odběrateli se zbožím a službami.
- Systém tři dny před uzavřením faktury oznámí oprávněnou osobu, že faktura bude končit.
- Na konci měsíce systém uzavře fakturu a uvědomí pracovníka. Oprávněná osoba fakturu vytiskne a pošle zákazníkovi.

- „Měsíční faktura“ je vždy vedena jako „Faktura se splatností“ a dokud k ní nebyla přiřazena platba je evidována jako „Neuhrazená“.

3.3.2.6 UC6 Evidence přijatých plateb

Aktéři: Oprávněná osoba, zákazník. **Cíl:** Evidovat platby k daným fakturám.

Vstupní podmínky: Faktura byla vydána a není uhrazena.

Tok událostí: UC6 začíná v momentu, kdy byl předán daňový doklad a zákazník hradí fakturu.

Scénář 1 – Přijetí hotovosti:

- Oprávněná osoba, k fakturování a přijímání hotovosti, v systému otevře formulář pro novou fakturu.
- Vybere zákazníka a systém vyplní hlavičku. Způsob úhrady zvolí „Hotově“.
- Naplní fakturu službami a zbožím, které chce fakturovat. Jednotlivé položky se, přidáním na fakturu, budou odečítat z evidence skladu.
- Ve formuláři nové faktury zvolí akci „Platit hotově“. Systém vytvoří platbu s variabilní symbolem dané faktury.
- Oprávněná osoba přijme hotovost a vytiskne fakturu. Systém eviduje fakturu jako „Faktura za hotové“ a jako „Uhrazenou“.

Scénář 2 – Platba převodem:

- Oprávněná osoba servisu otevře internetové bankovníctví v prohlížeči.
- V systému otevře formulář plateb.
- Do formuláře zadá číslo faktury, jejíž platbu chce zaevidovat.
- Daná osoba vyplní částku platby podle přijaté platby na bankovním účtu.
- Systém fakturu označí za „Uhrazenou“ pouze tehdy, jestliže součet částek v platbách s variabilním symbolem dané faktury se bude rovnat částce uvedené na faktuře.

Scénář 3 – Vzájemné započtení faktur:

- V případě, že odběratel je zároveň i dodavatel, mohou se pohledávky uhradit způsobem zápočtu, kdy se určitá výše pohledávek uhradí určitou výší závazků.
- Firma, jež chce pohledávky započíst do zápočtu, pošle zápočtový list, s tabulkou sald faktur, které se vzájemně započtou.
- V systému otevře formulář plateb. Do formuláře zadá číslo faktury, jejíž platbu chce zaevidovat.
- Daná osoba vyplní částku platby podle částky v tabulce sald v zápočtu.
- Jestliže záda v akci „zaplatit“ plnou sumu, faktura bude evidovaná jako „Uhrazená“ a jestli zadá pouze poměrnou část, faktura bude evidována jako „Neuhrazená“ a systém dopočítá zbytek nutný k plné úhradě.

3.3.2.7 UC7 Úschova kol

Aktéři: Pracovník servisu, zákazník. **Cíl:** Uschovat či vydat zákaznickova kola.

Vstupní podmínky: Zákazník nechá kola v úschově.

Tok událostí: UC7 začíná, když zákazník přijede přezout kola

Scénář 1 – Uschování kol:

- Pracovník servisu fyzicky uschová zákazníkovi kola či pneumatiky.
- Otevře sekci úschovy kol a vytvoří nový záznam.
- Systém otevře formulář, ve kterém pracovník servisu zadá: SPZ vozu, rozměr kol, sklad, kde jsou kola uložena, kolik kol uložil a sezónu (zimní x letní x celoroční).
- Pracovník servisu přiřadí úschovu danému zákazníkovi a uloží záznam.
- Systém uloží záznam s aktuálním datem.

Scénář 2 – Vydání úschovy:

- Pracovník servisu vydá kola z úschovy a vyhledá danou úschovu podle jména zákazníka v systému a vyvolá nad ní akci „Vydání úschovy“.
- Systém uloží úschovu jako „vydanou“ a uloží datum vydání.

- Úschovu je možné smazat pouze tehdy, když je systémem vedená jako vydaná a 2 roky se neobnovila.

3.3.2.8 UC8 Příjem zboží

Aktéři: Oprávněná osoba. **Cíl:** Přidat zboží na sklad.

Vstupní podmínky: Dodavatel doručil zboží, které oprávněná osoba objednala (UC3).

Tok událostí: UC8 začíná, když dodavatel předal příjmový doklad.

Scénář – Příjem zboží:

- Dodavatel dovezl objednané zboží na sklad.
- Oprávněná osoba převezme příjmový doklad. Otevře si formulář pro příjem zboží a přidá položky z příjmového dokladu.
- Do formuláře zadá: kód zboží, název, nákupní cenu a daň.
- Po vyplnění údajů příjemku uloží do systému. Systém položky z příjemky uloží do tabulky skladu.

3.3.2.9 UC9 Přidání nové služby

Aktéři: Oprávněná osoba. **Cíl:** Vytvořit novou položku služby.

Vstupní podmínky: Žádné.

Tok událostí: UC9 začíná, když servis začal poskytovat novou službu a chce ji zařadit do systému.

Scénář – Nová služba:

- Evidence poskytovaných služeb je jednoduchá tabulka, která slouží pro naplňování seznamů položek.
- Do téhle tabulky lze přidávat položky buďto obecně v systému v sestavě pro přehled poskytovaných služeb nebo rovnou při naplňování seznamů položek v dodacích listech a fakturách.
- Pracovník vyplní jednoduchý formulář, kde zadá název a cenu a zvolí akci „Přidat položku“. Položka se pak objeví v tabulce služeb a může být přidána do faktury.

3.3.2.10 UC10 Evidence zákazníků

Aktéři: Oprávněná osoba, zákazník. **Cíl:** Vytvořit nové portfolio zákazníka.

Vstupní podmínky: Stálý zákazník.

Tok událostí: UC10 začíná, když se zákazník vyjádří, že bude stálým zákazníkem.

Scénář – Nový zákazník:

- Pracovník servisu otevře formulář „Nový zákazník“ a systém zobrazí dialogové okénko s výběrem pro fyzickou nebo právnickou osobu.
- Systém zákazníkovi přidělí ID.
- Pracovník vyplní údaje o zákazníkovi (jméno, příjmení, ČOP nebo IČO, adresa, kontakt, bankovní spojení).
- Systém založí nový záznam o zákazníkovi. V náhledu na daného zákazníka budou záložky pro seznamy objednávek, dodacích listů, faktur, úschovy a aut.
- Pracovník přidá informace o zákaznické voze přes formulář „Přidat vůz“. Pracovník vyplní údaje o voze a jeho kolech.

3.3.3 Trasovací matice případů použití

V této matici se funkční požadavky trasují na případy použití, přičemž křížek označuje řešení požadavku daným případem použití (viz tabulka 1).

Trasovací matice	Případy použití									
Funkční požadavky	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10
R1	X	X	X							
R2				X						
R3					X					
R4						X				
R5				X				X	X	
R6							X			
R7										X

Tabulka 1 – Trasovací matice

3.3.4 Přidělování přístupu k systému

Přístup do systému bude fungovat tak, že se v databázi nadefinují účty, které budou mít přístup do systému a systém vyhodnotí, který účet má jaká práva a podle toho bude posílat stránky, na které mají přístup.

Jsou tyto typy účtů:

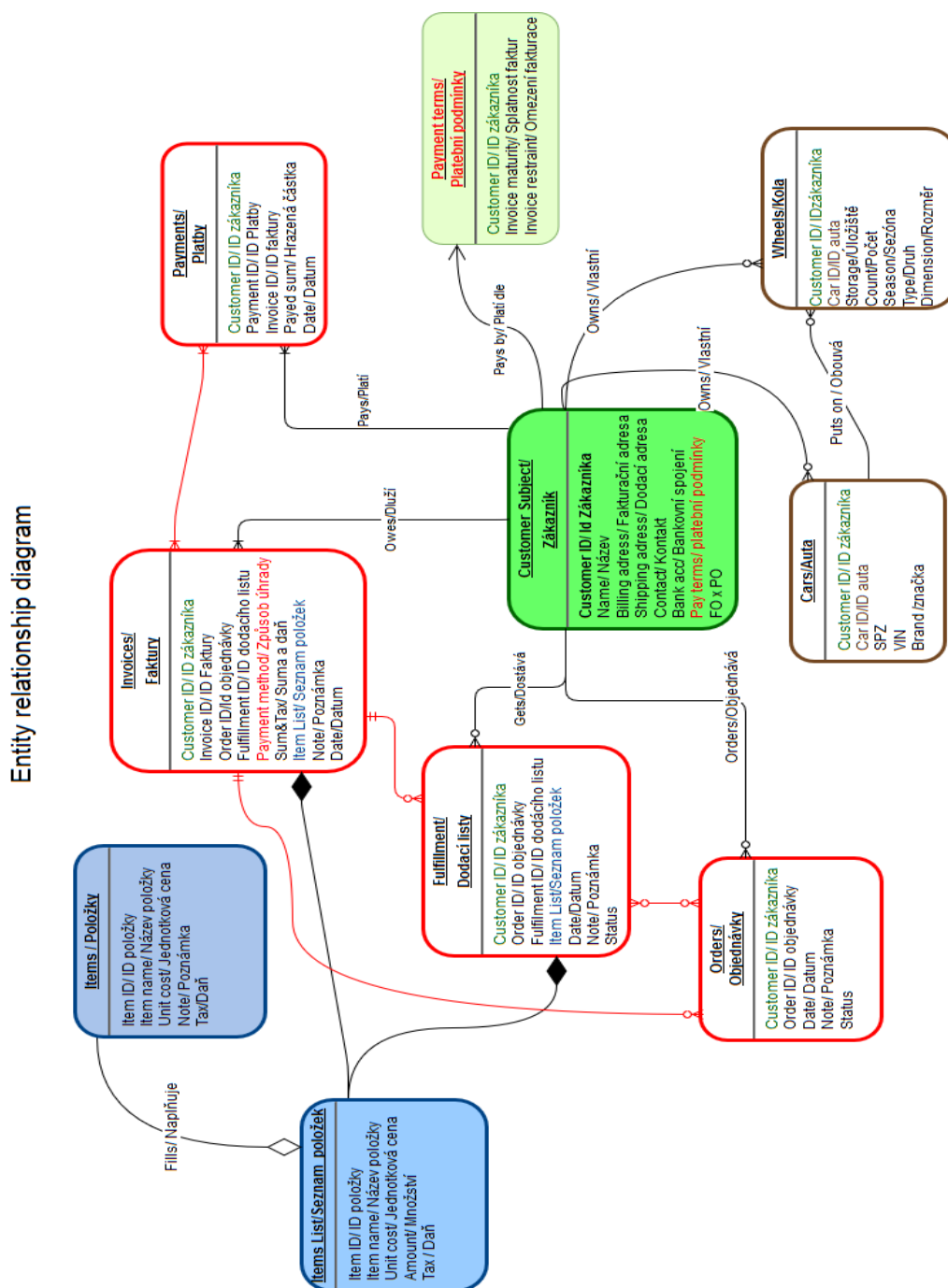
- Oprávněná osoba – neomezená práva
- Pracovník servisu – omezená práva pouze na určité úkony (viz 3.2.2 Případy použití)

4 Návrh iterace vývoje informačního systému

V této kapitole bude popsán návrh IS pomocí diagramů a databázové struktury.

4.1. Konceptuální model databáze systému

V této podkapitole popíšu jednotlivé entity, jejich atributy a vztahy mezi nimi. V tomto konceptuálním ER diagramu (obr.1) jsem uvedl atributy entit tak, aby tento diagram sloužil i jako předloha pro návrh databázové struktury.



Obrázek 2 - Konceptuální model databáze systému

4.1.1 Entita Zákazník

Tato entita reprezentuje zákaznicko portfolio v IS. Zákazníkovi se vedou záznamy o jeho autech, uschovaných kolech, objednávkách, dodacích listech, fakturách, platbách a platebních podmínkách.

Od této entity se odvíjejí hlavní vztahy.

- Zákazník vlastní auta a kola.
- Jeho autům se přezouvají kola.
- Zákazník objednává objednávky, dostává zboží s dodacím listem, dluží servisu za faktury a faktury platí podle platebních podmínek.

Zákazník, který se vysloví, že se chce stát stálým zákazníkem, musí poskytnout tyto informace:

- Jméno a příjmení nebo název společnosti
- ČOP nebo IČO – nutné pro ověření totožnosti.
- Adresa – nutné pro odesílání faktur či doručení zboží.
- Kontakt – pro komunikaci v případech konzultací problému.
- Bankovní spojení – pro přiřazení plateb k fakturám.

Tyto informace následně budou sloužit jako hlavička v zákaznicko portfolio a na všech dokladech, které k zákazníkovi patří.

Každý zákazníka má své vlastní platební podmínky podle zkušeností s ním a jeho platební morálkou.

Každý zákazník má ve svém portfolio své transakce, tj. objednávky, dodací listy, faktury a platby, které věcně naplňuje a servis s nimi pak dál operuje. Podle objednávek se naplňují dodací listy a faktury. Jak tyto transakce spějí do stádia, kdy jsou všechny vyřízeny, je potřeba je spojit se zákaznickou platbou. Jakmile je faktura uhrazená, transakce mezi servisem a zákazníkem je ukončená.

Pro zkvalitnění služeb je potřeba vést informace o zákaznických autech. Tyto informace budou v zákaznicko portfolio pro urychlení procesů komunikace se zákazníkem.

Dále v zákaznicko portfolio musí být informace o úschově, pokud využije této služby.

4.1.2 Entita Objednávky

Entita objednávky je reprezentací objednávek v systému, tedy začátek transakce. Objednávka vyjadřuje jednoduchý formulář, ve kterém je hlavička s údaji o zákazníkovi, který požaduje objednávku, a identifikační číslo objednávky s datem.

Předmět objednávky je prostý text, ve kterém je uvedeno, co se má provést, a seznam objednaných položek.

Objednávky mají vztah k dodacím listům, a to mnoho ku mnoha, protože jedna objednávka může být vyřízena více dodacími listy a naopak, tj. jeden dodací list může být vytvořen z více objednávek.

4.1.3 Entita Dodací listy

Tato entita reprezentuje dodací list, který je provázaný s objednávkami a zákazníkem. V hlavičce tohoto dokladu jsou informace o servise (dodavatel), o zákazníkovi (odběrateli), kterého se dodací list týká, o objednávkách, které tento doklad naplňuje a údaje samotného dodacího listu (číslo dokladu, název, datum a celková hodnota položek).

Předmětem je seznam položek, které jsou dodány. Tento seznam je tabulka s kódem, názvem a počtem kusů položky.

4.1.4 Entita Faktura

Tato entita reprezentuje daňový doklad. Je to klíčová transakce provázaná s objednávkami, dodacími listy, platbami a samozřejmě zákazníkem. Její hlavička vypadá podobně jako u dodacího listu. Hlavička obsahuje identifikační číslo faktury, objednávek a dodacích listů, které završuje. Na rozdíl od dodacího listu musí obsahovat informace bankovním spojení a o dani.

Předmětem faktury je seznam položek, které jsou fakturovány. Tento seznam je tabulka s kódem, názvem, jednotkovou cenou, počtem kusů, daní a součinem ceny a množství položky.

4.1.5 Entita Platby

Entita Platby slouží jako nástroj pro řízení pohledávek po zákaznících. Každá pohledávka může být placena více platbami a jedna platba může platit několik faktur. Podle zadané částky v platbě systém vyhodnotí, jestli je faktura uhrazená nebo ne.

4.1.6 Entita platební podmínky

Platební podmínky jsou reakcí na platební morálku zákazníka. Ve výchozím nastavení platebních podmínek bude mít každý zákazník 14dní splatnost faktur a může odebírat zboží a služby do hodnoty 50 tisíc korun českých.

V případě, že zákazník má špatnou platební morálku, lze toto nastavení změnit a daný zákazník bude muset platit pouze hotovostí. To ovšem závisí na rozhodnutí oprávněné osoby.

4.1.7 Entita Auta

Tato entita hraje klíčovou roli při objednávkách zákazníka, protože reprezentuje záznamy o jeho autech s informacemi pro objednání správných dílů a příslušenství k danému vozu. A zákazník nemusí hledat dokumenty o autě při každé návštěvě servisu.

4.1.8 Entita Kola

Pro servis jsou důležité pouze tyto informace: Značka auta, motorizace, státní poznávací značka, a hlavně kód VIN, podle kterého se vyhledávají, v dalších softwarových aplikacích, technické informace o vozidle.

4.1.9 Entity Položky a Seznam položek

Položky naplňují seznam položek, který je součástí entit faktury a dodací listy. Atributy entity položky vyjadřují základní informace položek na skladě, a to: identifikační číslo, název, cena, poznámka a příslušná daň.

4.2. Návrh logické struktury databáze

Databázová struktura je tvořena tabulkami, jejich sloupci a vazbami mezi nimi pomocí primárních a cizích klíčů. V této podkapitole bude popsány jednotlivé tabulky návrhu databázové struktury. Parametry relací logického relačního modelu jsou zachyceny na obrázku 3.

4.2.1 Tabulka customer

Hlavní tabulkou je tabulka **customer**, ve které se ukládají data o zákazníkovi. Tyto data zákazníka se pak využívají v hlavičkách objednávek, dodacích listů a faktur.

K tabulce **customer** jsou připojeny tabulky **cars**, reprezentující auta a **wheels**, reprezentující kola v úschově, vztahem jedna ku mnoha, protože jeden zákazník může mít více aut.

Dále tabulka **customer** je spojena vazbou jedna ku mnoha s tabulkami **order**, **fulfill**, **invoice** a **payment** (tabulky reprezentující objednávky, dodací listy, faktury a platby).

4.2.2 Tabulky transakcí

Tabulky transakcí jsou tabulky:

- Invoice – pro faktury.
- Fulfill – pro dodací listy.
- Order – pro objednávky.
- Payment – pro platby.

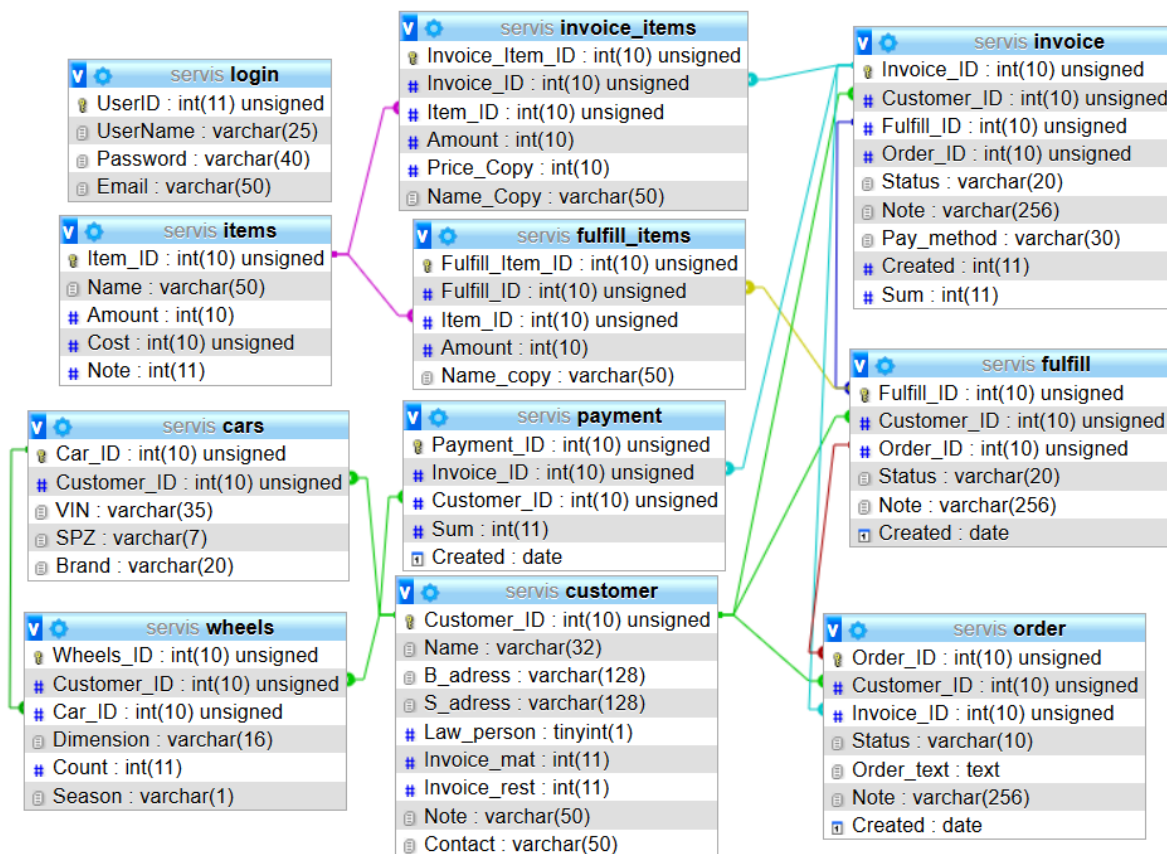
Vazby tabulky **invoice** s ostatními tabulkami transakcí jsou všechny jedna ku mnoha, tudíž jedna faktura může mít více objednávek, dodacích listů a plateb.

Tabulka **fulfill** má vztah k tabulce **order** jedna ku mnoha, protože jeden dodací list může být vytvořen z více objednávek.

K tabulkám **invoice** a **fulfill** jsou připojeny vztahem mnoho ku jedné tabulky **invoice_items** a **fulfill_items**, které slouží pro uložení jednotlivých seznamů položek ve fakturách a dodacích listech. K těmto tabulkám je připojena tabulka **items** vztahem mnoho ku jedné, ve které se ukládají položky na skladě.

4.2.3 Tabulka login

Osamocená tabulka login bez žádných vazeb slouží čistě k přihlašování do systému. Její atributy jsou: identifikační číslo uživatele, jméno, heslo a email. Email slouží pro obnovení hesla a může sloužit také k upozornění. Heslo je uloženo jako zašifrovaný řetězec a systém bude porovnávat pouze tyto zašifrované řetězce.



Obrázek 3 – Logická struktura databáze

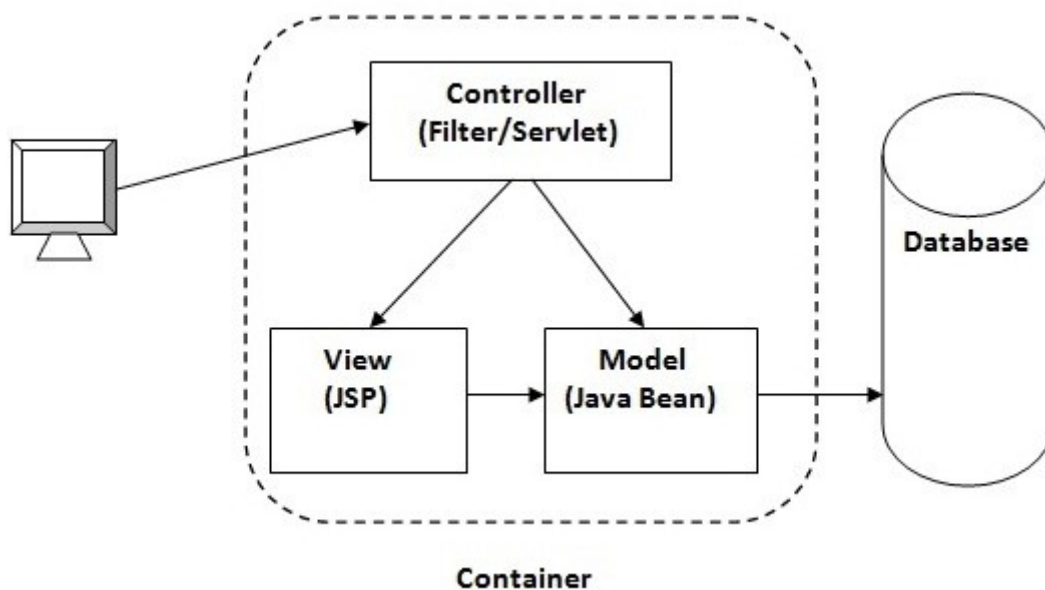
5 Tvorba a testování prototypu

Jako prototyp do této práce jsem se rozhodl vytvořit interaktivní grafické uživatelské rozhraní s částečnou funkcionalitou pomocí HTML5, CSS3 a JavaScriptu.

Přes uživatelské rozhraní tohoto systému bude uživatel posílat požadavky na server, kde program napsaný v Javě vyhodnotí požadavek a pošle na stranu klienta požadovanou stránku s požadovaným obsahem. Program na straně serveru bude komunikovat s databází pomocí SQL příkazů a dotazů (viz obrázek 4).

Pro specifikaci návrhu řešené serverové části jsem využil principu návrhového vzoru MVC.

„Model“ je komponenta, která představuje jednotlivé třídy se strukturou atributů, odpovídající struktuře tabulek v databázi a s těmi pak aplikace pracuje při změnách dat, výběrech dat atp. skrze komponentu „Controller“. Výsledky operací se pak zobrazují skrze "View" prvky, což jsou pak např. webové stránky.



Obrázek 4 – Návrhový vzor MVC

Grafické uživatelské rozhraní část rozhraní použiji jako prezentaci zákazníkovi (servisu), ve které zákazník zjistí, jestli je to produkt, který požadoval, a jestli má intuitivní a jednoduchou ovladatelnost, anebo je potřeba něco úplně změnit, lehce upravit, něco přidat či ubrat. V přílohách 1-10 jsou snímky obrazovek prototypu.

Nejprve jsem vytvořil HTML strukturu stránky podle návrhu grafického uživatelského rozhraní. Využil jsem Grid systému z knihovny Bootstrap, abych rozmístil základní komponenty a tím základní rozvržení stránky. Čtyři hlavní komponenty stránky jsou navigační panel, menu, samotný obsah stránky a patička.

Navigační panel slouží pro navigaci na stránce. Menu rozděluje stránku na sekce podle obsahu a kontextu. Menu je umístěno na levé straně. Obsahová část slouží pro poskytování informací a je umístěna uprostřed stránky. Patička nese informace o stránce a ukončuje obsah stránky a je umístěna ve spodní části.

Bootstrap světově oblíbený framework pro responsivní a mobile-first stránky. Tento framework nabízí mnoho šablon komponent a různých elementů pro vytváření struktury stránky i obsahové části stránky. Bootstrap pro většinu komponent používá JavaScript, aby fungovali. [10]

Bootstrap Grid system je výkonný nástroj pro sestavení layoutu stránek všech tvarů a velikostí, díky dvanáctisloupcovému systému, pěti výchozích responsivních úrovní, Sass proměnných a mnoho předdefinovanými funkcemi. [10]

Bootstrap grid system používá řadu obalů, řádků, sloupců pro rozvržení a zarovnání obsahu. [10]

Cílem této platformy je zlepšení přístupnosti dat v HTML tabulkách. Pro dosažení tohoto cíle, jsou uživatelé dělení do dvou skupin: vývojáři, kteří využívají tento nástroj pro tvorbu webových aplikací, a koncoví uživatelé, kteří používají toto rozhraní v prohlížeči.

V těchto tabulkách je uživatel schopen jednoduše třídit, řadit a hledat informace. [9]

Technologie C3 Chart komfortní, přizpůsobitelná a kontrolovatelná. Funguje pro jednoduché generování grafů, založených na D3 kódu, obalením požadovaného kódu ke konstrukci celého grafu. [10]

Při generování, C3 poskytuje třídy pro každý element, takže je možné definovat si styl podle potřeby pomocí třídy a je možné dál strukturu rozšiřovat přímo pomocí D3. [10]

C3 poskytuje variace API pro přistupování k stavu grafu i po vykonání. [10]

5.1. Návrh grafické uživatelského rozhraní

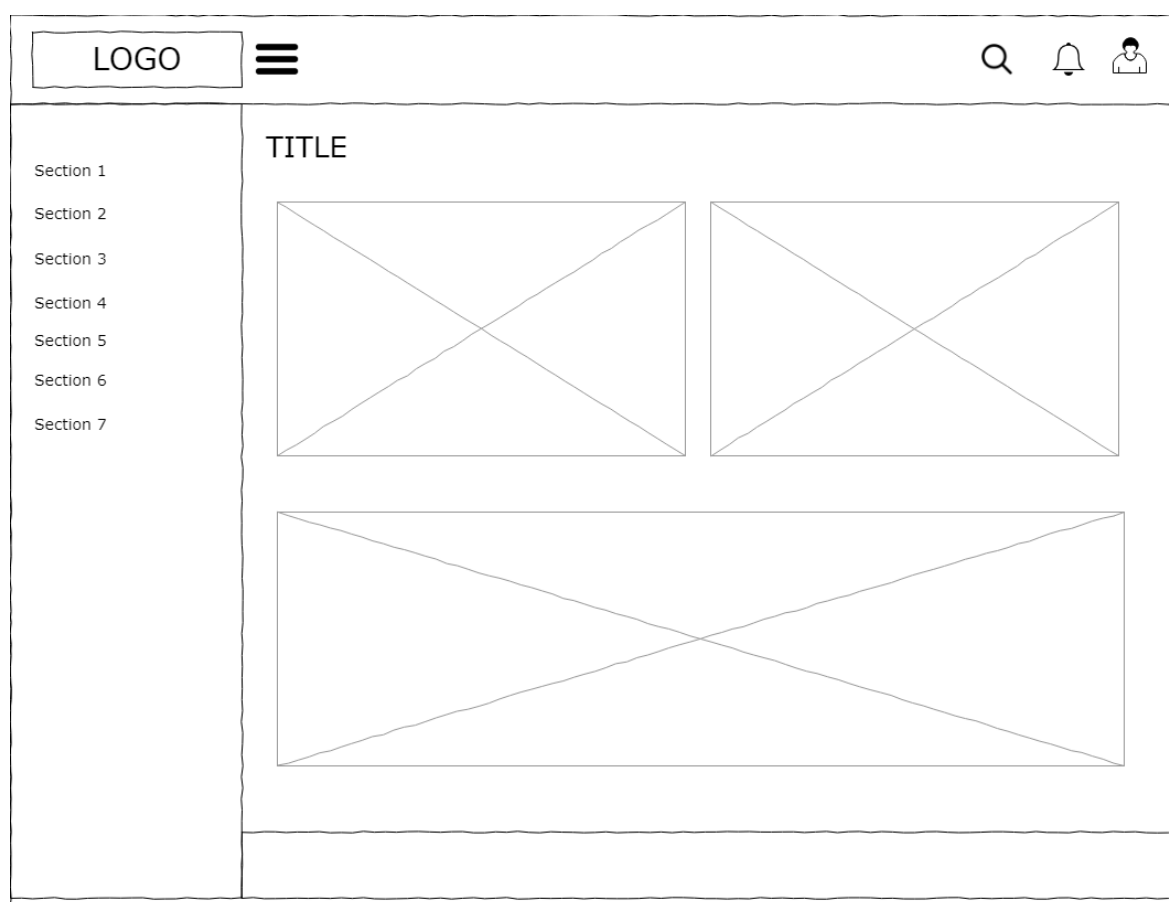
O návrhu jsem uvažoval jako o jednoduchém a přehledném rozložení stránky. Tento návrh slouží jako kostra všech dalších navazujících stránek.

Navigační panel je nahoře přes celou šířku stránky, kde vlevo je logo servisu a vpravo je vyhledávací panel, s upozorněními a nastavením přihlášeného účtu.

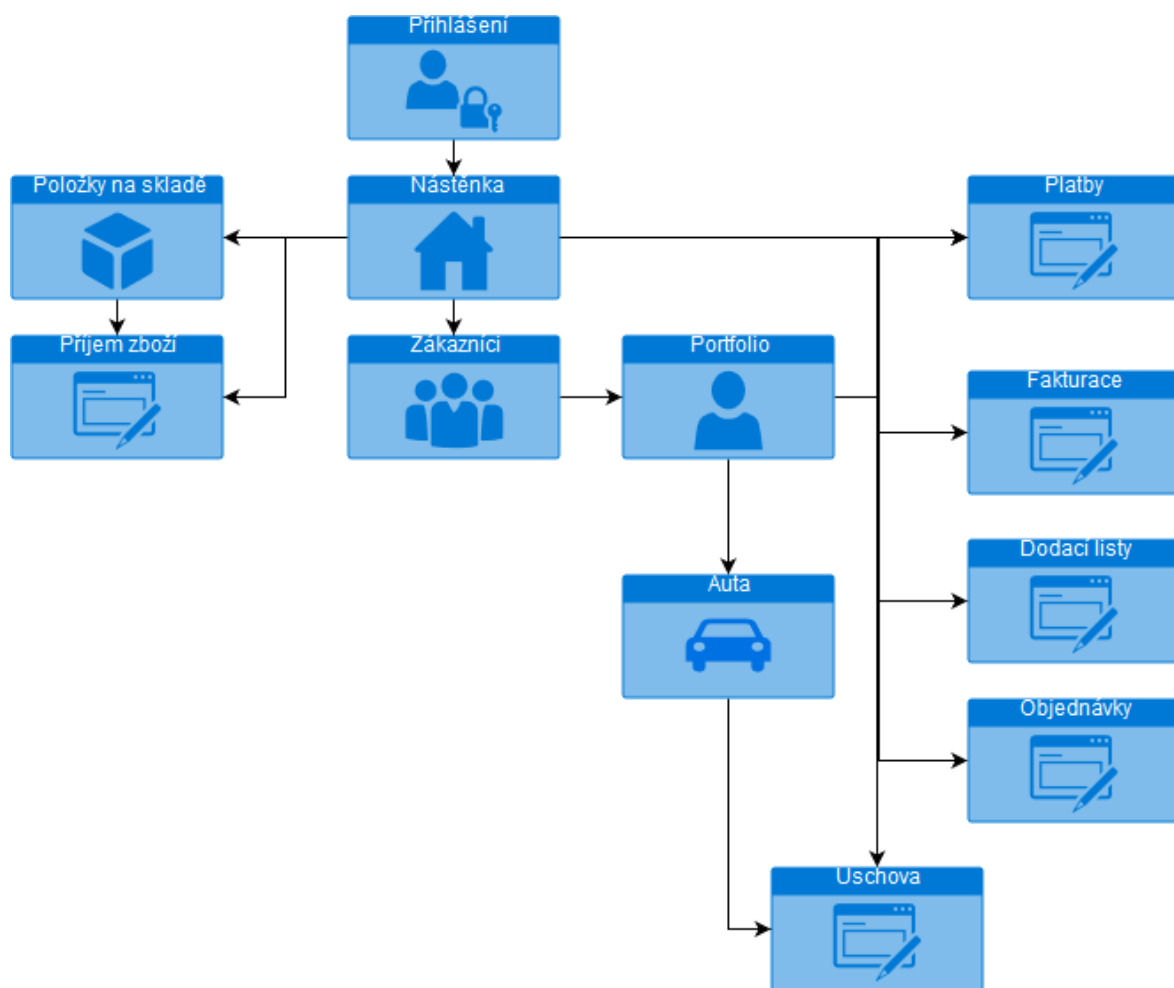
Menu je umístěno v levém boku stránky, které je otevíratelné pomocí tlačítka v navigaci. V menu jsou jednotlivé sekce systému, které mohou být rozevírací v případě, že se sekce dělí na více částí.

Obsahová část je tvořena responsivními okénky, ve kterých se zobrazují obsahové elementy stránky.

Ve spodní části stránky se nachází patička, ve které jsou umístěny informace o stránce a systému.



Obrázek 5 – Rozvržení prvků rozhraní klientské části systému



Obrázek 6 - Stromový diagram stránek webové aplikace

Na obrázku 6 je graficky znázorněna navigační struktura webové aplikace.

Jako první se zobrazí přihlašovací stránka, kde uživatel zadá přihlašovací údaje. Po přihlášení bude přesměrován na úvodní stránku, ze které se dostane na jakoukoliv stránku aplikace potřebuje pomocí menu. Z nástěnky lze přejít na stránky plateb, fakturace, dodacích listů, objednávek a úschovy. Na těchto stránkách jsou přehledy a tlačítko pro otevření formuláře podle sekce.

Ze stránky seznamu zákazníků lze přejít, kliknutím na danou kartu zákazníka, do zákaznickova portfolia, kde jsou přehledy zákaznickových transakcí a úschov.

5.2. Hlavní prvky rozhraní (stránky) webové aplikace

Hlavní stránky webové aplikace jsou:

Nástěnka – úvodní stránka webové aplikace

Zákazníci – seznam zákazníků se základními informacemi

Portfolio zákazníka – stránka s příslušnými tabulkami zákazníka

5.2.1 Nástěnka

Úvodní stránka neboli nástěnka, je první stránka, která se zobrazí po přihlášení uživatele do systému. Na stránku jsem umístil různé grafy s užitím technologie C3 Chart a pomocí JavaScriptu nástěnku s úlohami (tzv. „TODO“).

Grafy jsou pouze orientační a pro ilustraci jsem jako předmět grafů použil denní, měsíční a roční příjmy.

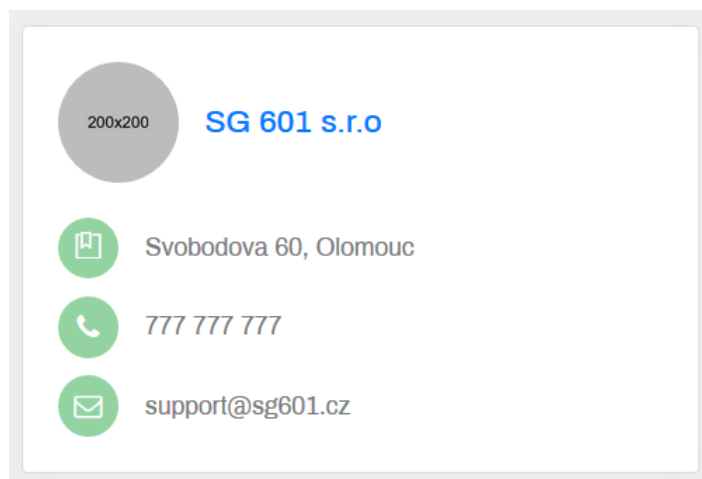
Dále by na nástěnce mohli být novinky. Přehled nově uhrazených faktur, vyřízených objednávek. Viz příloha 1.

5.2.2 Zákazníci

Na stránce v obsahové části se nachází kartičky karet reprezentující jednotlivé zákazníky. Nahoře v navigaci je vyhledávání, které slouží pro vyhledávání jakéhokoliv zákazníka. Viz příloha 2.

Zákazníková karta obsahuje základní informace o zákazníkovi. Při kliknutí na zákaznickou kartu se otevře stránka, kde je více informací o zákazníkovi. Díky technologii Bootstrap je celá stránka responsivní a velikost kartiček se odvíjí od velikosti okna prohlížeče.

Kartička je tvořena komponentou Bootstrapu Card, která je flexibilní a roztažitelný obal. Obsahuje možnosti pro hlavičku i patičku, široké spektrum obsahu a plno zobrazovacích možností.



Obrázek 7 - Karta zákazníka na seznamu zákazníků

5.2.3 Portfolio zákazníka

Portfolio zákazníka je také tvořeno karty. Každá karta má jiný obsah. Obsahová stránka dokumentu portfolio.html začíná kartou se základními informacemi o zákazníkovi jako: název, adresa, telefon, IČO a email. Viz přílohy 3 a 4.

Následuje karta Faktury, na které je zobrazena část tabulky seřazené podle nejnovější. Na této kartě jsou dvě tlačítka: jedno pro zobrazení celé tabulky faktur a druhá pro vytvoření nové faktury. Při kliknutí do řádku se otevře daná faktura v režimu pro čtení. Pokud fakturace dané faktury není dokončena, otevře se v režimu úprav pro přidání položek či dodacích listů.

Pod kartou faktur je karta s platebními podmínkami, kde je uvedeno kolik dní má daný zákazník splatnost faktur a do jaké částky může odebírat zboží na fakturu. Na této kartě jsou dvě tlačítka: jedno pro změnu těchto podmínek a druhé pro návrat do výchozího stavu.

Dalšími kartami v portfolio jsou karta objednávek a karta dodacích listů, které fungují obdobně jako karta faktur.

Karta úschovy zobrazuje tabulku úschov. Je možno přidat novou úschovu či vydat úschovu.

Poslední karta je karta aut, která obsahuje tabulku s auty a jejich informacemi. Opět možno přidat nové či zobrazit celou tabulku.

6 Závěr

Cílem této práce bylo navrhnout a vytvořit prototyp systému pro autoservis. Tento systém bude sloužit pro správu informací o zákaznících a jejich příslušenství a mimo bude umožňovat autoservisu spravovat evidenci položek na skladě, faktur, dodacích listů, objednávek a plateb. Tím přináší přidanou hodnotu autoservisu, hlavně v oblasti řízení vztahů se zákazníkem, protože ulehčuje procesům komunikace.

Kapitola „Teoretická východiska tvorby databázových aplikací“ se zabývala především základními předpoklady a seznámení s technologiemi potřebnými pro navrzení a vytvoření plné verze tohoto systému.

Kapitola „Analýza současného stavu a definice požadavků“ se zabývala popisem autoservisu a jeho stávajících procesů a popisem základních požadavků systému

V této práci jsem navrhnul strukturu prototypu systému pro autoservis a vytvořil částečně funkční prototyp v podobě webové aplikace s grafickým uživatelským rozhraním. Tento prototyp slouží pro prezentaci zákazníkovi této práce, tedy autoservisu. Zadavatel se věcně vyjádří, co se mu líbí, co by chtěl změnit, co přidat a co odebrat.

V kapitole „Návrh iterace vývoje informačního systému“ jsou popsány a zobrazeny navrhované objekty databáze aplikace a vazby mezi nimi, které je potřeba při programování zohlednit, aby měl systém věcnou logiku. Dále byl popsán a zobrazen návrh logické struktury databáze pro databázi tohoto systému.

V kapitole „Tvorba a testování prototypu“ je nastíněn postup vytváření prototypu, návrh grafického uživatelského rozhraní a popisy hlavních webových stránek. Tato kapitola obsahuje použité technologie a popis jejich využití.

Do budoucna bych chtěl, aby tento systém dokázal synchronizovat platby s internetovým bankovníctvím. Dále bych chtěl rozšířit jeho funkcionalitu o účetnictví, statistiky účetnictví a ekonomické ukazatele. Chtěl bych vylepšit správu zákazníků tak, aby systém zasílal upomínky neplaticím zákazníkům v určitém časové intervalu. Dalším rozšířením bych uvažoval správu zaměstnanců a jejich výkonů.

Seznam použité literatury

Monografické publikace

1. GRÄSSLE, P.; BAUMANN, H.; BAUMANN, P. *UML 2. 0 in Action: A Project-based Tutorial*; Packt Publishing: Birmingham, 2005
2. KROENKE, David a David J. AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.
3. CONNOLLY, T., BEGG, C. *Database Systems: A Practical Approach to Design, Implementation, and Management, Global Edition, 6th Edition*. Pearson, 2014. ISBN 9781292061184.
4. PECINOVSKÝ, Rudolf. *Java 9: kompletní příručka jazyka*. Praha: Grada Publishing, 2018. Knihovna programátora (Grada). ISBN 978-80-271-0715-5.
5. CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 9788025137338.

Internetové zdroje

6. MySQL 5.7 Reference Manual :: 1.3.1 What is MySQL?. MySQL :: Developer Zone [online]. Copyright © 2019, Oracle Corporation and [cit. 05.05.2019]. Dostupné z: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
7. Cloud ERP Software Solution | NetSuite. [online]. Copyright © Oracle [cit. 06.05.2019]. Dostupné z: <http://www.netsuite.com/portal/products/erp.shtml>
8. About JavaScript - JavaScript | MDN. [online]. Copyright © 2005 [cit. 05.05.2019]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript
9. Reference. DataTables | Table plug-in for jQuery [online]. Copyright © 2007 [cit. 05.05.2019]. Dostupné z: <https://datatables.net/reference/index>
10. Bootstrap · The most popular HTML, CSS, and JS library in the world. [online]. Dostupné z: <https://getbootstrap.com/>

11. What is UML | Unified Modeling Language. Welcome To UML Web Site!
[online]. Copyright © 1997 [cit. 05.05.2019]. Dostupné z:
<https://www.uml.org/what-is-uml.htm>
12. C3.js | D3-based reusable chart library. C3.js | D3-based reusable chart library
[online]. Dostupné z: <https://c3js.org/>

Seznam zkratek

API	Application Programming Interface
BLOB	Binary Large Object
CPU	Central processing unit
CSS	Cascading Style Sheets
CRM	Customer Relationship Management
ČOP	Číslo občanského průkazu
DLL	Data Definition Language
DML	Data Manipulation Language
ER	Entity Relationship
ERP	Enterprise Resource Planning
GUI	Graphical user interface
HTML	HyperText Markup Language
IČO	Identifikační číslo obchodníka
IS	Informační systém
OMG	Object Management Group
SŘBD	Systém řízení báze dat
SQL	Structured Query Language
UML	Unified Modeling Language
UC	Use Case

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová (bakalářská) práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že bibliografické údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou (bakalářskou) práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 10. května 2019

.....

Filip Skulník

Seznam příloh

Příloha č. 1: Nástěnka webové aplikace

Příloha č. 2: Seznam zákazníků

Příloha č. 3: Portfolio zákazníka první část

Příloha č. 4: Portfolio zákazníka druhá část

Příloha č. 5: Formulář pro přidání zákazníka první část

Příloha č. 6: Formulář pro přidání zákazníka druhá část

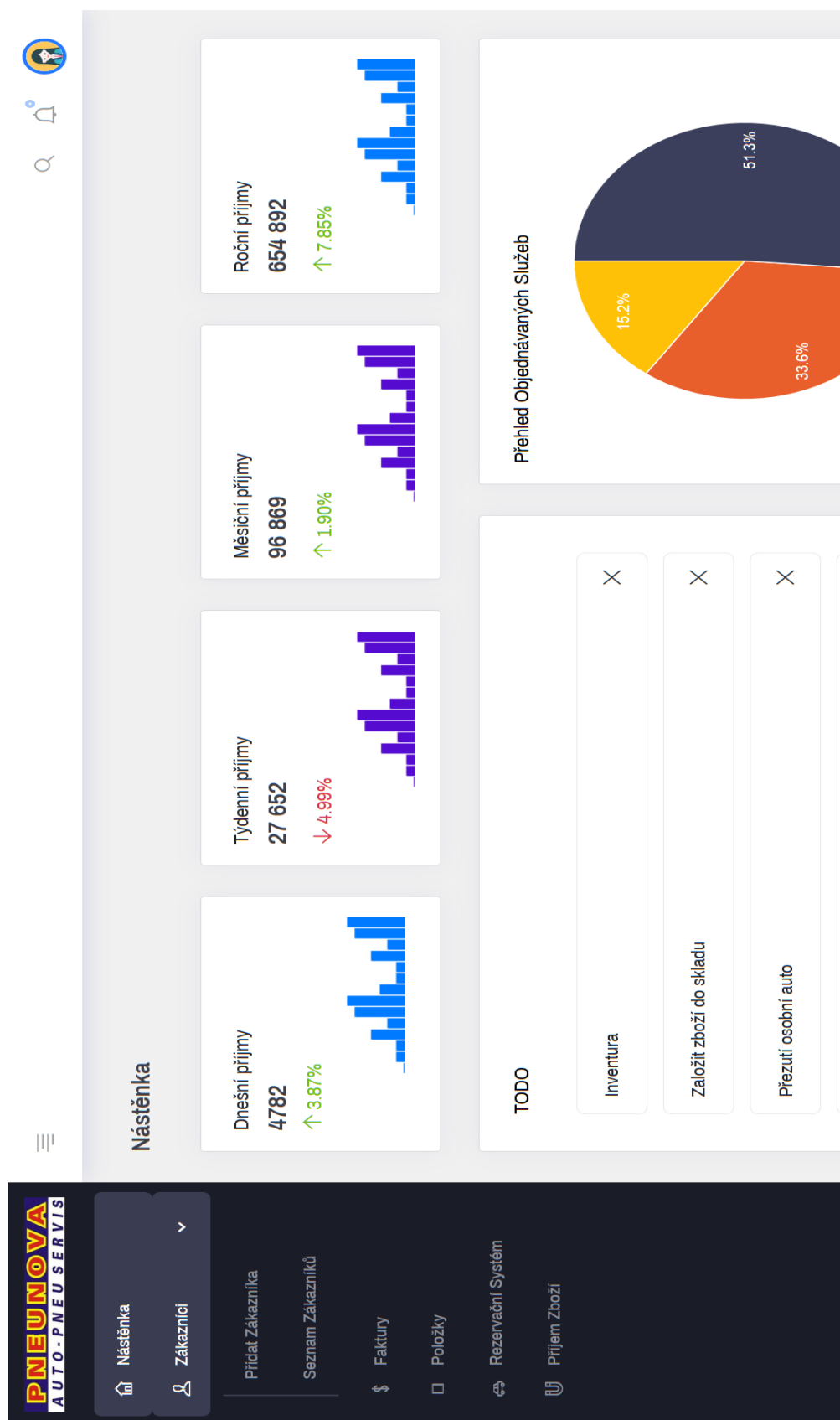
Příloha č. 7: Formulář pro příjem zboží

Příloha č. 8: Přehled zboží

Příloha č. 9: Přehled služeb

Příloha č. 10: Přehled faktur

Příloha č. 1: Nástěnka webové aplikace



Příloha č. 2: Seznam zákazníků

Příloha č. 3: Portfolio zákazníka první část

400x400

SG 601 s.r.o.

Sokolská 15, Olomouc

732 680 745

i IČO:56894712

support@sg601.cz

Faktury Zákazníka

Číslo faktury	Firma	Datum vystavení	Splatnost	Částka
123456	SG 601 s.r.o.	23.9.2016	16.5.2018	4670 Kč
123456	SG 601 s.r.o.	23.9.2018	24.10.2019	1250 Kč

Zobrazit více

Nová faktura

Platební Podmínky

Splatnost faktur: 14 dní

Omezení fakturace: do 50000 Kč

Upravit

Výchozí nastavení

51

Příloha č. 4: Portfolio zákazníka druhá část

Objednávky Zákazníka

Číslo objednávky	Datum vystavení
123456	27.2.2018
123456	8.3.2018

Zobrazit víceNová objednávka

Dodací Listy Zákazníka

Číslo dodacího listu	Datum vystavení
123456	27.2.2018
123456	8.3.2018

Zobrazit víceNový dodací list

Úschovy Zákazníka

Číslo úschovy	Datum úschovy	Datum vydání
123456	27.2.2018	
123456	8.3.2018	8.4.2018


Zobrazit vícePřidat úschovuVydát úschovu


Auta Zákazníka


Číslo auta	Značka	Model	Motorizace	Rok výroby	VIN Kód
123456	Audi	A4	2.0 TDI	2008	WVGZZZ1TZDW048829
123456	BMW	3	325d	2006	WVGZZZ1TZDW048829


Zobrazit víceNové auto


Příloha č. 5: Formulář pro přidání zákazníka první část





 Nástěnka


 Zákazníci


 Přidat Zákazníka


 Seznam Zákazníků

 Faktury


 Položky


 Rezervační Systém

 Příjem Zboží



HLAVNÍ INFORMACE






 Nahrát obrázek


JMÉNO

PŘÍJMENÍ

FIRMA



Příloha č. 6: Formulář pro přidání zákazníka druhá část



Nástěnka

Zákazníci

Přidat Zákazníka





Seznam Zákazníků

Faktury

Položky

Rezervační Systém

Příjem Zboží



KONTAKT

ADRESA

MĚSTO

TELEFONÍ ČÍSLO

IČO

EMAIL


Uložit







Pneunova s.r.o. ©




Příloha č. 7: Formulář pro příjem zboží

Příjem Zboží	
Číslo příjemky	
Skladník	
Číslo skladu	
Položka	
Množství	
Nahrát fakturu / dodací list	
Nahrát dokument	Nahrát

Příloha č. 8: Přehled zboží



 Nástěnka
  Zákazníci
  Faktury
  **Položky**
 Rezervační Systém
  Příjem Zboží

Identifikační číslo	Název	Cena bez DPH (Kč)	Cena s DPH (Kč)	Množství na skladě
P01265	Pneumatika Continental 135/45/17	1936,- Kč	2450,- Kč	12
P01269	Pneumatika Michelin 135/45/17	1817,- Kč	2300,- Kč	8
P01200	Barum Brilliantis 2 165/70/14	702,- Kč	889,- Kč	4
P12380	Pirelli W 240 SottoZero II 285/35/19	3152,- Kč	3990,- Kč	8
P18853	Markum NKT Sport 165/65/14	990,- Kč	1198,- Kč	16
P18163	Seiberling Touring 165/65/14	990,- Kč	1198,- Kč	6
P18111	Seiberling Touring 165/65/14	1990,- Kč	2408,- Kč	4
P18504	Barum Brilliantis 165/65/14	1300,- Kč	1573,- Kč	8

Položky

Příloha č. 9: Přehled služeb



Nástěnka

Zákazníci

Faktury

Položky

Rezervační Systém

Přijem Zboží

P18111	Seiberling Touring 165/65/14	1990,- Kč	2408,- Kč	4
P18504	Barum Brilliantis 165/65/14	1300,- Kč	1573,- Kč	8

Název	Cena s DPH
Kompletní přezutí 4ks plechová kola 13" a 14"	450,- Kč
Kompletní přezutí 4ks plechová kola 15" až 17"	490,- Kč
Kompletní přezutí 4ks Alu kola 13"	590,- Kč
Kompletní přezutí 4ks Alu kola 17"	690,- Kč
Kompletní přezutí 4ks Alu kola 19"	790,- Kč
Kompletní přezutí 4ks dodávky a SUV Alu	690,- Kč
Geometrie přední nápravy	700,- Kč

Pneunova s.r.o ©

Příloha č. 10: Přehled faktur

Faktury

COPY

CSV

EXCEL

PDF

PRINT

ZOBRAZ ZÁZNAMŮ

25

▼

HLEDAT:

ZOBRAZUJI 1 AŽ 6 Z CELKEM 6 ZÁZNAMŮ

Číslo faktury	Firma	Datum vystavení	Splatnost	Částka
123456	Tomášek Radim	23.9.2016	16.5.2018	4670 Kč
123456	Tomášek Radim	23.9.2018	24.10.2019	1250 Kč
123456	ORSA realitní kancelář	2.2.2019	2.5.2018	1647 Kč
123456	Mechanika družstvo	13.2.2016	13.2.2017	17456 Kč
123456	PV Auto s.r.o.	21.4.2017	21.5.2017	16521 Kč
123456	Cars Tech s.r.o.	21.6.2018	21.7.2018	12721 Kč

Předchozí

1

Další